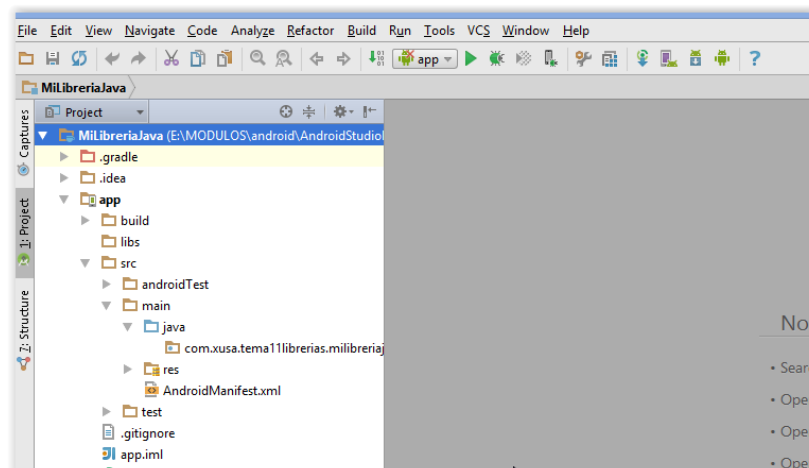


COMO CREAR E IMPORTAR LIBRERÍAS EN ANDROID STUDIO

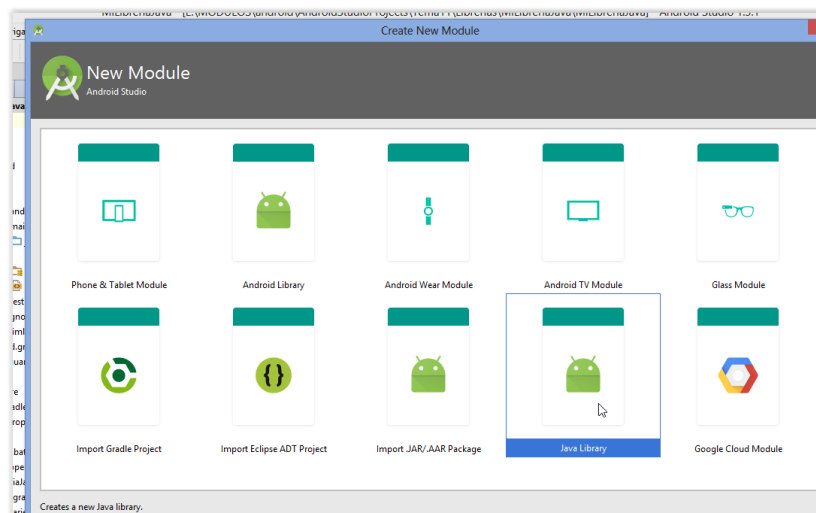
1. COMO CREAR UNA LIBRERÍA .JAR EN ANDROID STUDIO

Para crear una librería .jar en Android Studio, deberemos añadir un módulo nuevo e indicar que debe ser una librería de java. A ese módulo podemos añadirle todas las clases que necesitemos (copiándolas de otro proyecto o creándolas nuevas). Después le diremos al Gradle que compile y cree el .jar. Paso a paso lo de debemos hacer será lo siguiente:

1. Puedes crear primero un proyecto Android de forma normal, pero sin activity principal.

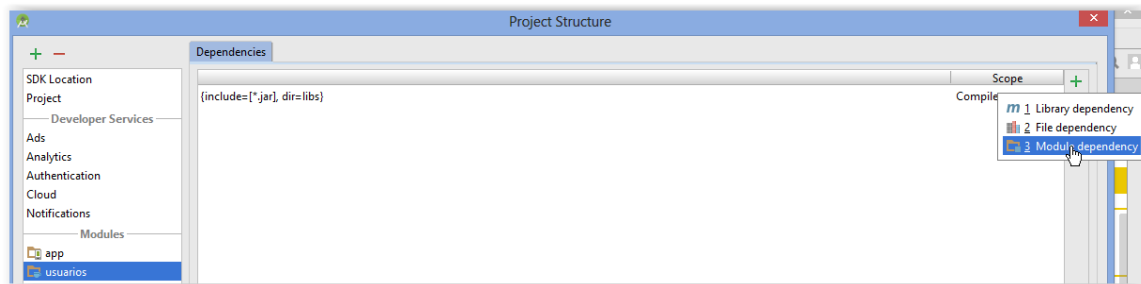


2. Añadiremos el módulo Java, File -> New Module -> Java Library. Indica el nombre del paquete y de módulo. La clase se creará por defecto, pero posteriormente podrás eliminarla, si no la necesitas.

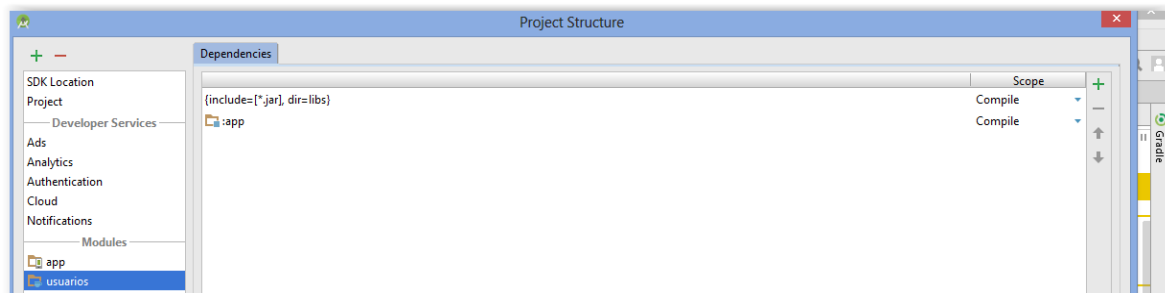


3. Una vez hecho esto, se habrá creado un nuevo módulo con el nombre que le hayamos indicado. A ese módulo le podemos añadir las clases que necesitemos para nuestra librería. Como será un módulo Java no podremos añadir clases que tengan dependencias con librerías de Android.

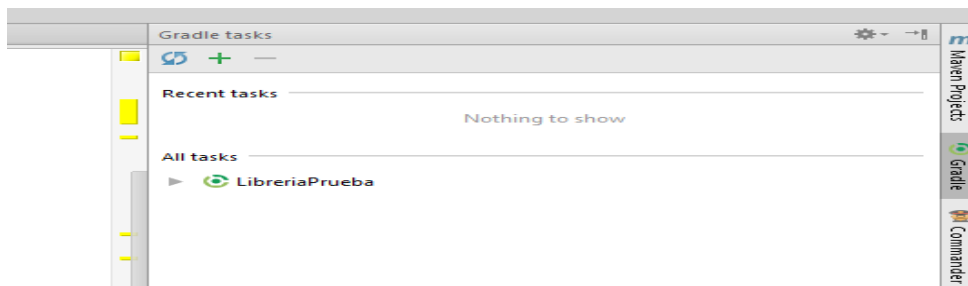
4. El siguiente paso, será crear una dependencia entre ambos módulos. Esto lo haremos desde la ventana Project structure, en el módulo java le añadiremos la dependencia del módulo Android. Ver siguiente captura de pantalla.



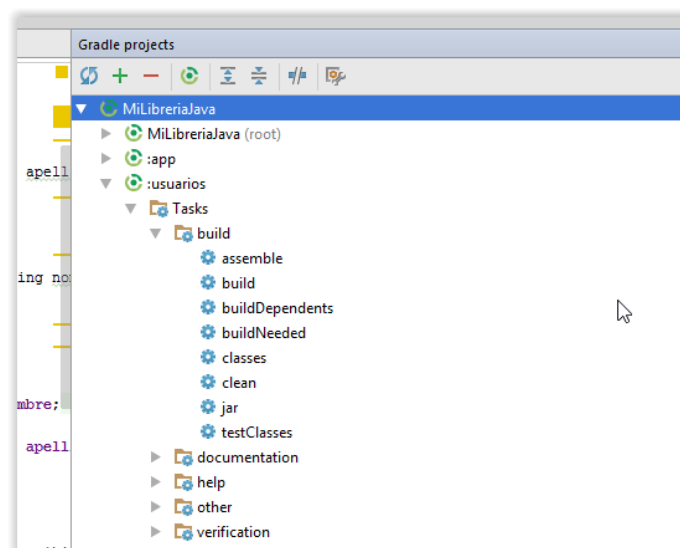
Una vez hecho quedará así:



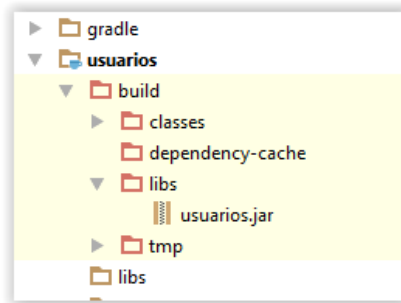
5. Después de esto quedará, solamente, crear el .jar. Para ello primero abriremos la ventana de Gradle Tasks:



6. Sobre el modulo que hemos creado, desplegamos y seleccionamos el Jar:



7. Esto nos creará el .jar que buscábamos y que posteriormente podremos añadir a los proyectos en que nos haga falta. El .jar lo crea en el directorio build->libs del módulo creado.



✚ Crea una librería Java siguiendo los anteriores pasos (el punto jar). Deberá llamarse MiLibreriaJava, estará en el paquete com.tunombre.usuarios y deberá tener al menos las dos clases de las cuales os pasamos el código.

```
public class Usuarios {
    String nombre;
    String apellidos;
    Direccion direccion;

    public Usuarios(String nombre, String apellidos) {
        this.nombre = nombre;
        this.apellidos = apellidos;
    }

    public Usuarios(String apellidos, String nombre, Direccion direccion) {
        this.apellidos = apellidos;
        this.nombre = nombre;
        this.direccion = direccion;
    }

    public String getNombre() { return nombre; }

    public String getApellidos() { return apellidos; }

    @Override
    public String toString() {
        return "Usuarios{" +
            "apellidos='" + apellidos + '\'' + ", nombre='" + nombre + '\'' +
            ", direccion=" + direccion + '}';
    }
}

public class Direccion {
    String calle;
    short numero,piso;
    String letra, poblacion, provincia;
    int cpostal;

    public Direccion(String calle, short numero, String poblacion, int cpostal, String provincia) {
        this.calle = calle;
        this.numero = numero;
        this.poblacion = poblacion;
        this.cpostal = cpostal;
        this.provincia = provincia;
    }

    public Direccion(String calle, String provincia, String poblacion, short piso, short numero,
        String letra, int cpostal) {
        this.calle = calle;
        this.provincia = provincia;
        this.poblacion = poblacion;
        this.piso = piso;
        this.numero = numero;
        this.letra = letra;
        this.cpostal = cpostal;
    }

    public String getCalle() {return calle;}

    public String getProvincia() {return provincia;}

    public String getPoblacion() {return poblacion;}

    public short getPiso() {return piso;}

    public short getNumero() {return numero;}

    public String getLetra() { return letra;}

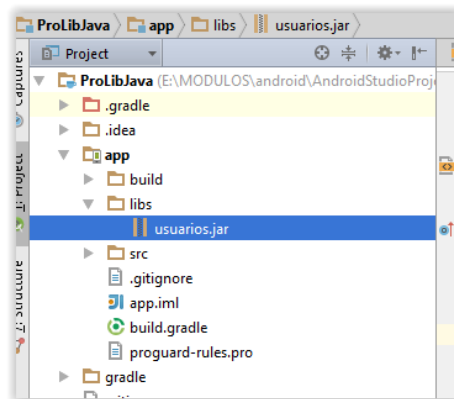
    public int getCpostal() {return cpostal;}

    @Override
    public String toString() {
        return "Direccion{" + "calle='" + calle + '\'' + ", numero=" + numero +
            ", piso=" + piso + ", letra='" + letra + '\'' +
            ", poblacion='" + poblacion + '\'' + ", provincia='" + provincia + '\'' +
            ", cpostal=" + cpostal + '}';
    }
}
```

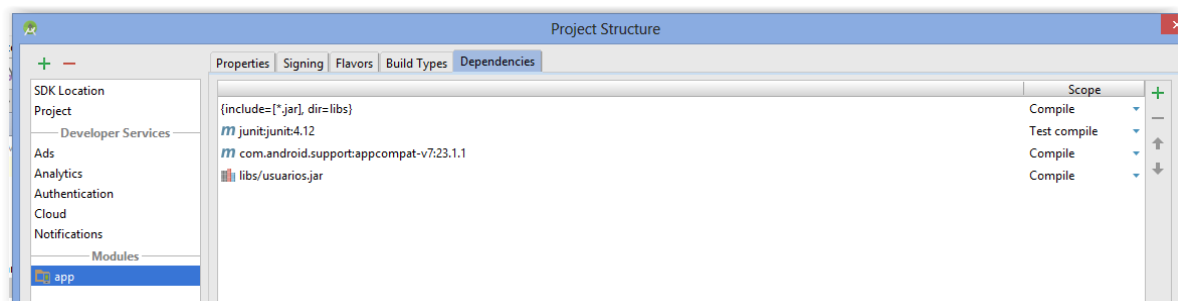
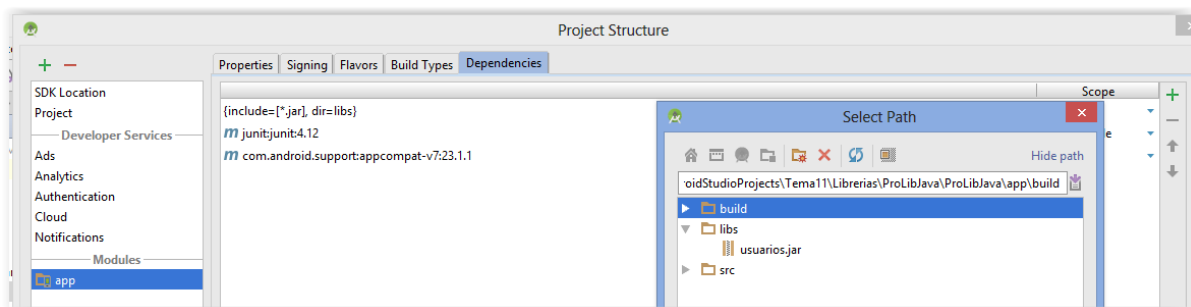
2. COMO AÑADIR UNA LIBRERÍA .JAR EN ANDROID STUDIO

Los pasos a seguir para añadir un .jar externo a nuestra aplicación, son muy sencillos:

1. Copiamos el archivo .jar en la carpeta lib o en la carpeta build de nuestro proyecto.



2. Iremos a Project structure-> app -> Dependencies -> + -> File Dependencies y añadiremos el fichero que hemos copiado en libs o build (en este caso en libs)



3. Con solo esto ya podemos importar el paquete y usar sus miembros públicos.

```
import com.xusa.librerias.Direccion;
import com.xusa.librerias.Usuarios;

public class MainActivity extends AppCompatActivity {
    TextView datos;
    Short n=22;

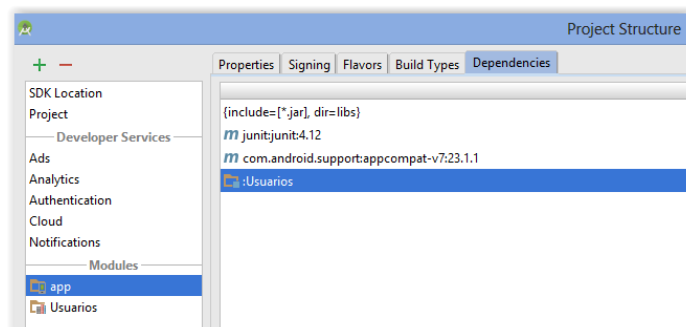
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        datos=(TextView) findViewById(R.id.datos);
        Direccion direccion=new Direccion("la calle",n,"alicante",0333,"Alicante");
        datos.setText(new Usuarios ("Pep","Gar",direccion).toString());
    }
}
```

- ✚ Crea un proyecto al que le añadas la librería que hemos implementado en el ejercicio anterior y prueba su funcionamiento con un código parecido al de la imagen

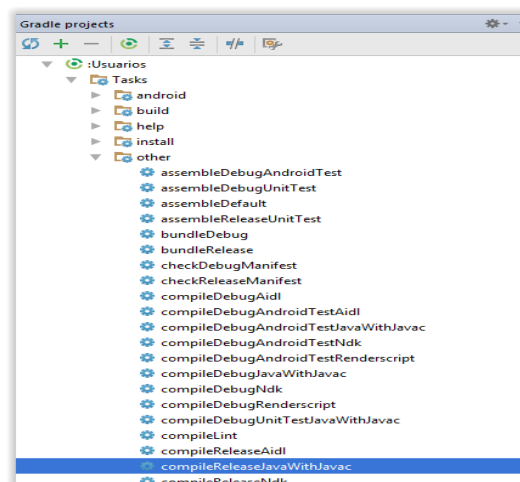
3. COMO CREAR UNA LIBRERÍA .AAR EN ANDROID STUDIO

La librería .aar es específica de Android studio, por tanto, sí nos permite exportar cualquier tipo de elemento sea de Java o sea de Android.

1. Primero deberemos crear un proyecto de la manera acostumbrada, podremos indicarle que sea vacía la actividad o posteriormente borrarla si no necesitamos incluir nada en ella.
2. Deberemos crear un nuevo módulo, que será una **librería de Android**. Le pondremos, a nuestra conveniencia, el espacio de nombres y demás elementos que se nos pida añadiendo la activity o no. Podremos copiar en este nuevo módulo las clases que necesitemos o crearlas en el directamente.
3. El siguiente paso, será crear una dependencia entre ambos módulos. Esto lo haremos desde la ventana Project structure, en este caso la dependencia será desde el módulo principal al de la librería (es decir, nos creamos una dependencia en la app del proyecto del módulo añadido).



4. Una vez hecho lo anterior, iremos al Gradle Tasks and seleccionaremos, del módulo que hemos añadido en el paso anterior, la opción de `CompileReleaseJava`, como podéis ver en la imagen. Esto generará el .aar del módulo especificado y lo podremos encontrar en **directorio_del_modulo\build\outputs\aar**.



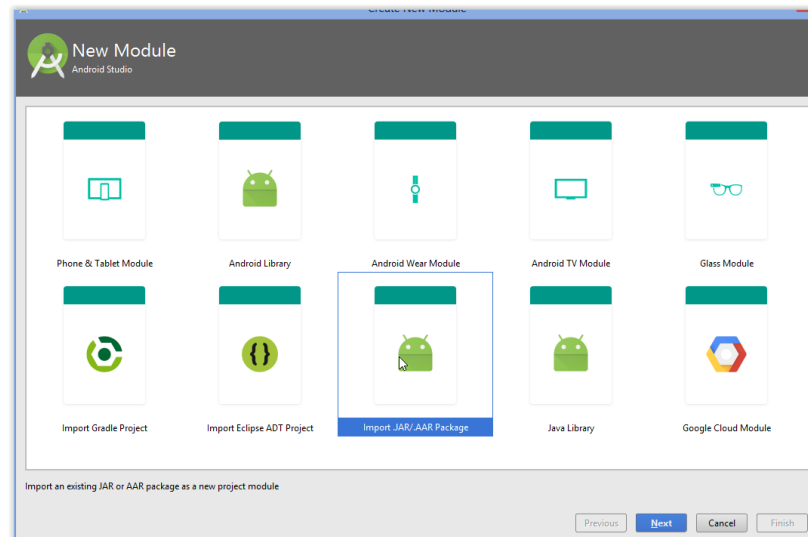
Con estos pasos tendremos el .aar construido y preparado para ser importado por otro proyecto.

- ✚ Crea la misma librería usuarios que hemos implementado para los ejercicios anteriores, pero ahora como una librería de Android.

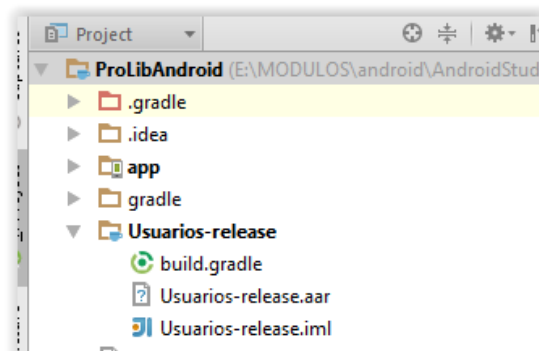
4. COMO AÑADIR UNA LIBRERÍA .AAR EN ANDROID STUDIO

Y si lo que necesitamos es reutilizar una librería .aar en otra aplicación, deberemos seguir los siguientes pasos:

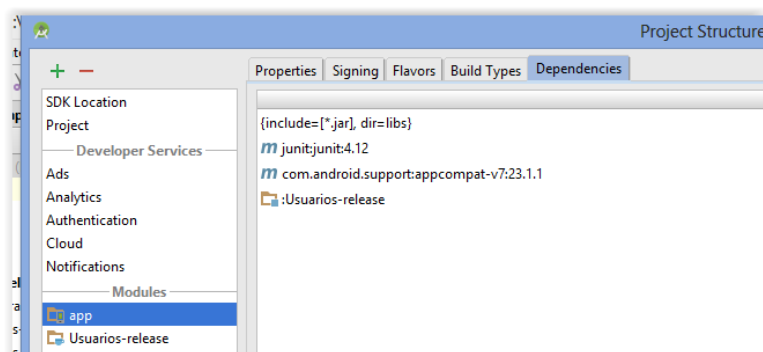
1. En la aplicación donde queremos añadirlo, deberemos crear un nuevo módulo seleccionando la opción de import Jar o AAR package. Indicando la ruta donde se ha guardado el .aar.



Veremos que se ha creado un módulo similar al siguiente:



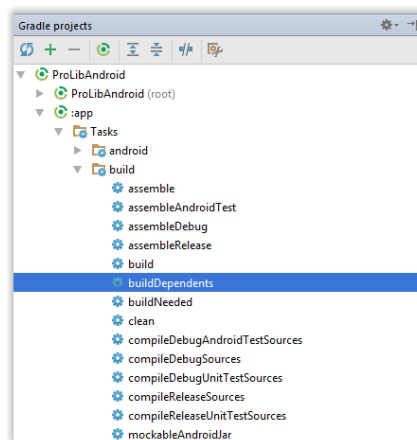
2. En project structure añadiremos la dependencia del módulo de la aplicación con el nuevo creado.



Esto nos modificará el Build.Gradle con la segunda línea de la captura siguiente:

```
dependencies {  
    compile fileTree(include: ['*.jar'], dir: 'libs')  
    testCompile 'junit:junit:4.12'  
    compile 'com.android.support:appcompat-v7:23.1.1'  
    compile project(':Usuarios-release')  
}
```

Para terminar deberemos indicarle en el Gradle Tasks que cree las dependencias del modulo (buildDependencies), y con esto en un principio se habrá añadido todo lo necesario para que nuestro proyecto pueda funcionar con la librería .aar añadida.



- ✚ Crea un proyecto al que siguiendo los pasos anteriormente explicados, le añadas la librería Android, Usuarios, que hemos implementado en el ejercicio anterior y prueba su funcionamiento.